 ISO/IEC JTC 1/SC 29/WG 2 N0226

**ISO/IEC JTC 1/SC 29/WG 2  
MPEG Technical requirements   
Convenorship: SFS (Finland)**

**Document type:** Output Document

**Title:** Requirements for render-based systems for MPEG Phase 1 V1.0

**Status:** Approved

**Date of document:** 2022-07-22

**Source:** ISO/IEC JTC 1/SC 29/WG 2

# Expected action: None

# Action due date: None

**No. of pages:** 6 (with cover page)

**Email of Convenor:** igor.curcio@nokia.com

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg2>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 2**

**MPEG TECHNICAL REQUIREMENTS**

**ISO/IEC JTC 1/SC 29/WG 2 N** **0226**

**Online - July 2022**

|  |  |
| --- | --- |
| **Title** | **Requirements for render-based systems for MPEG Phase 1 V1.0** |
| **Source** | **WG 2, MPEG Technical requirements** |
| **Status** | **Approved** |
| **Serial Number** | **21829** |

**Introduction**

A set of current and emerging immersive displays are characterized by “families” according to the optical technologies used by each display to create each display’s optimal visual experience [1]. This document provides requirements for MPEG to develop one or more specifications that collectively describe a “system” to deliver media to a render-based client that may be attached to an immersive display as described in [1], where such media is contained within an ITMF container [2]. Furthermore, the ITMF container is planned to be extended by MPEG extensions to include support for VPCC, GPCC, MPEG audio, and MIV. Such extensions also form a part of the “system.” Additionally, ITMF may be extended to include compression technologies of media types that are natively supported by ITMF, including but not limited to VDB, Alembic, FBX, OBJ, and USD.

An initial deliverable for the “system” is identified as MPEG-I Part 28. Other parts are expected to be developed as work to develop the “system” progresses in future MPEG meetings. A separate document [3] provides draft requirements that remain to be addressed as part of the “system,” i.e., that are not defined herein.

ED Note: The requirements in this document are not final at this version of the document, and additional inputs are expected and solicited.

**Glossary**

***Scene graph***

general data structure commonly used by vector-based graphics editing applications and modern computer games, which arranges the logical and often (but not necessarily) spatial representation of a graphical scene; a collection of nodes and vertices in a graph structure.

***Node***

fundamental element of the scene graph comprised of information related to the logical or spatial or temporal representation of visual or audio information;

***Attribute***

metadata associated with a node used to describe a particular characteristic or feature of a node either in a canonical or more complex form (e.g. in terms of another node).

.

***Container***

a serialized format to store and exchange information to represent all natural, all synthetic, or a mixture of synthetic and natural scenes including a scene graph and all of the media resources that are required for rendering of the scene

[***Serialization***](https://en.wikipedia.org/wiki/Serialization)

the process of translating [data structures](https://en.wikipedia.org/wiki/Data_structure) or [object](https://en.wikipedia.org/wiki/Object_(computer_science)) state into a format that can be stored (for example, in a [file](https://en.wikipedia.org/wiki/Computer_file) or memory [buffer](https://en.wikipedia.org/wiki/Data_buffer)) or transmitted (for example, across a [network](https://en.wikipedia.org/wiki/Computer_network) connection link) and reconstructed later (possibly in a different computer environment). When the resulting series of bits is reread according to the serialization format, it can be used to create a semantically identical clone of the original object.

***[Renderer](https://en.wikipedia.org/wiki/Rendering_(computer_graphics))***

a software-based) application or process, based on a selective mixture of disciplines related to: acoustic physics, [light physics](https://en.wikipedia.org/wiki/Optics), [visual perception](https://en.wikipedia.org/wiki/Visual_system), audio perception, [mathematics](https://en.wikipedia.org/wiki/Mathematics), and [software development](https://en.wikipedia.org/wiki/Software_engineering), that, given an input scene graph and asset container, emits a visual and/or audio signal suitable for presentation on a targeted device or conforming to the desired properties as specified by attributes of a render target node in the scene graph. For visual-based media assets, a renderer may emit a visual signal suitable for a targeted display, or for storage as an intermediate asset (e.g. repackaged into another container i.e. used in a series of rendering processes in a graphics pipeline); for audio-based media assets, a renderer may emit an audio signal for presentation in a multi-channel loudspeaker and/or binauralized headphones, or for repackaging into another (output) container. Examples of renderers include: Octane, Redshift, Renderman, V-Ray, and real-time visual and audio renderers used in Unity, and Unreal.

***Scripting language***

An interpreted programming language that can be executed by a renderer at runtime to process dynamic input and variable state changes made to the scene graph nodes, which affect rendering and evaluation of spatial and temporal object topology (including physical forces, constraints, kinematics, deformation, collisions), and energy propagation and transport (light, sound).

***Shader***

a type of [computer program](https://en.wikipedia.org/wiki/Computer_program) that was originally used for [shading](https://en.wikipedia.org/wiki/Shading) (the production of appropriate levels of [light](https://en.wikipedia.org/wiki/Light), [darkness](https://en.wikipedia.org/wiki/Darkness), and [color](https://en.wikipedia.org/wiki/Color) within an image) but which now performs a variety of specialized functions in various fields of computer graphics [special effects](https://en.wikipedia.org/wiki/Special_effects) or does [video post-processing](https://en.wikipedia.org/wiki/Video_post-processing) unrelated to shading, or even functions unrelated to graphics at all.

***Ray tracing***

refers to a type of technique for 3D rendering that faithfully simulates real-world optical effects and is capable of producing photorealistic digital content.

***Rasterization***

refers to a type of technique for 3D rendering that maps 3D representations of scenes to a 2D output buffer, and is commonly used for most real-time applications (e.g., games). Most modern GPUs are highly optimized for rasterization in addition to ray tracing and deep learning.

***scene-based media***

audio, visual, haptic, and other primary types of media and media-related information organized logically and spatially by a use of a **scene graph**

Note to entry: **scene-based media** may be further organized into a linear series of scenes for time-based presentations, or into a branch-based structure for interactive presentations.

***frame-based media***

2D video with or without associated audio

***Node-graph***

Node structure (usually a node tree) that describes an asset, e.g., Alembic, glTF, FBX.

**Functions to be defined with respect to Part 28**

Part 28 shall perform the following essential functions:

* Defining an independent mapping space addressing multiple presentation engines, e.g., Unity and Unreal Engines.
* Defining a mapping between the independent mapping space and at least one presentation engine, e.g. Unreal Engine.
* Defining ITMF extensions to MPEG media types, e.g., MIV, VPCC, GPCC, MPEG Audio.

**MPEG requirements pertaining to MPEG-I Part 28**

* The specification shall define an independent mapping space (IMS) that addresses the scene graphs between ITMF and multiple presentation engines including Unreal Engine and Unity Engine, using the most recent Long Term Support version of each engine.
* The specification shall define an architecture for streaming scene-based media to Unreal and Unity Engine clients.
* The IMS shall include the following features
  + Scene graph structure: The specification shall define
    - Terms that describe the structure and organization of information and media assets in a scene graph, e.g., nodes, node descriptors, inputs, and outputs of each node
    - Semantics for connections that exist between nodes
    - Semantics for nodes that are organized into groups.
  + Scene structure: The specification shall define
    - Minimum elements needed in the IMS to represent an ITMF scene
    - Elements of an ITMF scene that may assume a default value
    - Minimum elements to create an Unreal Engine scene representation
    - Elements of an Unreal Engine scene representation that may assume a default value.
    - Minimum elements to create a Unity scene representation.
    - Elements of a Unity scene representation that may assume a default value.
  + Node graphs: The specification shall define
    - The semantics of an independent node graph
    - The minimal structure of an independent node graph
    - The semantics of incorporating an independent node graph into a scene
  + Data types: The specification shall define:
    - All data types that are defined by ITMF.
    - Data types that are defined by Unreal Engine, but are not defined by ITMF.
    - Data types that are defined by Unity Engine, but are not defined by Unreal Engine nor ITMF.
    - Semantics for processing each data type in the IMS.
  + Geometry: The specification shall define:
    - The types of geometric assets that are supported in the ITMF
    - The types of geometric assets that are supported by Unreal Engine but are not supported by the ITMF.
    - The types of geometric assets that are supported by Unity Engine but are not supported either by the ITMF nor by Unreal Engine.
    - Semantics for how metadata provided by each geometric asset are incorporated or merged in with metadata that may be supplied by a scene graph that references each geometric object.
  + Node types: The specification shall define:
    - All node types that are defined by ITMF.
    - Node types that are defined by Unreal Engine, but are not defined by ITMF.
    - Node types that are defined by Unity Engine, but are not defined either by ITMF nor Unreal Engine.
    - The semantics of each node type relative to rendering the scene by a presentation engine.
* The specification shall define extensions to the ITMF using the scripted node mechanism of ITMF.
  + The specification shall define an ITMF scripted node that enables the carriage of a V-PCC geometry asset in an ITMF scene graph. This scripted node shall serve as an informative portion to the specification.
  + The specification shall define an ITMF scripted node that enables the carriage of a G-PCC geometry asset in an ITMF scene graph. This scripted node shall serve as an informative portion to the specification.
  + The specification shall define an ITMF scripted node that enables the carriage of a MIV video asset in an ITMF scene graph. This scripted node shall serve as an informative portion to the specification.
* The specification shall define a mapping between the independent mapping space and at least one presentation engine, e.g., Unreal Engine. The mapping shall serve as an informative portion to the specification.

1. **References**
2. “Draft report on framework for characterizing emerging immersive displays and media for immersive applications,” ISO/IEC SC29/WG2 Output Document Serial Number M55595, October 2020.
3. [ITMF specifications](https://www.immersivealliance.org/download/download-itmf/).
4. “Revised Draft requirements for render-based systems beyond MPEG Phase 1” m60473, MPEG 139, online, July 2022.